DOCUMENT RESUME

ED 076 304 RC 007 015

AUTHOR	Block, Karen K.; And Others
TITLE	A Computer Resource for the Elementary School. Progress Report 1971-1972.
INSTITUTION	Pittsburgh Univ., Pa. Learning Research and Development Center.
SPONS AGENCY	National Inst. of Education (DHEW), Washington, D.C.; National Science Foundation, Washington, D.C.
PUB DATE	72
NOTE	48p.
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Assisted Instruction; *Elementary Schools; *Individualized Instruction; Mathematics Instruction; Problem Solving; *Small Schools; Spelling Instruction; *Technology; Testing Programs

ABSTRACT

The appropriate and effective uses of the computer in an individualized school were examined in this project. The project investigated the uses of computers for instructional assistance, including testing, management, tutorial instruction, drill and practice, exploration and discovery, and problem solving; the design of a small computer time-sharing system with a multi-language capability, and the processes underlying developmental work in computer-assisted instruction. A small mobile computer system was used at the Oakleaf Elementary School in Baldwin-Whitehall, Pennsylvania, to test the program components; computer system design; computer-assisted testing; computer-assisted instruction in spelling, mathematics, and problem solving techniques; and computer-assisted management. Plans for 1972-73 were presented for each area, with considerations for determining the computer's potential. (PS)

LEARNING RESEARCH AND DEVELOPMENT CENTER

1973/1

A COMPUTER RESOURCE FOR THE ELEMENTARY SCHOOL: PROGRESS REPORT 1971 - 1972

KAREN K. BLOCK, MARTHENA CARLSON, ROBERT J. FITZHUGH, TSE-CHI HSU, ERIC [illegible]COBSON, GILBERT FUENTE, RICHARD A. ROMAN, JEROME ROSNER, DOROTHEA P. SIMON, ROBERT GLASER, AND WILLIAM W. COOLEY

RECEIVED JUL 5 1973 NMSU E.R.I.C.

FILMED FROM BEST AVAILABLE COPY

ED 076304

RC 007015

ED 076304

RECEIVED
JUL 5 1973
NMSU
E.R.I.C.

A COMPUTER RESOURCE FOR THE ELEMENTARY SCHOOL

PROGRESS REPORT 1971 - 1972

Karen K. Block
Marthena Carlson
Robert J. Fitzhugh
Tse-Chi Hsu
Eric Jacobson
Gilbert Puente
Richard A. Roman
Jerome Rosner
Dorothea P. Simon

Robert Glaser and William W. Cooley
Principal Investigators

Learning Research and Development Center
University of Pittsburgh

Summer 1972

The research reported herein was supported by a grant from the National Science Foundation (NSF-GJ-540X) and by the Learning Research and Development Center, supported in part as a research and development center by funds from the National Institute of Education (NIE), United States Department of Health, Education, and Welfare. The opinions expressed in this publication do not necessarily reflect the position or policy of the sponsoring agencies and no official endorsement should be inferred.

TABLE OF CONTENTS

INTRODUCTION 1

COMPUTER SYSTEM DESIGN 4

COMPUTER-ASSISTED TESTING 9

COMPUTER-ASSISTED INSTRUCTION IN SPELLING 15

COMPUTER-ASSISTED PRACTICE FOR MATHEMATICS SKILLS 23

COMPUTER-ASSISTED INSTRUCTION IN PROBLEM-SOLVING TECHNIQUES 31

COMPUTER-ASSISTED MANAGEMENT OF INSTRUCTION 35

USING A COMPUTER IN AN ELEMENTARY SCHOOL: QUESTIONS FOR STUDY 38

REFERENCES 44

INTRODUCTION

The Learning Research and Development Center (LRDC) at the University of Pittsburgh is a university-wide research institute concerned with the design and study of instructional systems. In particular, it is concerned with environments for learning that are adaptive to the individual characteristics of students. Prototype systems developed and evaluated at the Center over the past eight years have been widely adopted by school systems throughout the country, and related products are being made available by education industries. Along side of and integral to its development and evaluation efforts is the Center's program of basic research oriented toward understanding the cognitive processes of young children and the environmental conditions which influence learning during the elementary-school years.

As part of this work, LRDC has carried out a series of hardware, software, and instructional studies on possibilities for the use of computers in elementary schools. Since 1969, with the sponsorship of the National Science Foundation, the Center has conducted a project to integrate computer assistance into an instructional school model explicitly designed for an individualized, adaptive educational program. The basic purpose of the project is to investigate appropriate and effective uses of the computer in an individualized school in order to foster the adoption of individualized systems of elementary education at a time when individualization is much talked about, but inordinately difficult to implement in the traditional school setting (see, for example, Glaser & Cooley, in press).

In particular, the project is concerned with: (1) investigation of the many uses of computers for instructional assistance, including testing, management, tutorial instruction, drill and practice, exploration and discovery, and problem solving; (2) the design of a small computer time-sharing system with a multi-language capability, which can be used for both operations and research and which can eventually lead to the development of specifications for a small computer system especially suited for elementary-school use; and (3) investigation of the basic processes underlying developmental work in computer-assisted instruction (CAI), including the psychological characteristics of elementary-school students and of instruction in problem solving and inquiry, the appropriate timing and distribution of use of computerized practice exercises, and the decision-making characteristics of computer testing.

Following initial design and development work and pilot experimentation on computer hardware, software, and educational components, a small mobile computer system was introduced into the Oakleaf Elementary School in the Pittsburgh suburb of Baldwin-Whitehall in the middle of the 1971-72 school year. Oakleaf, a small school consisting of 9 teachers and 226 children, is totally committed to an individualized setting for elementary education from kindergarten through fifth grade, and has been run as a cooperative effort between the school system and LRDC since 1964. It is important to emphasize that the objective of LRDC's work at the school is not the development of isolated curriculum materials, but rather the design of a total instructional environment.

To date, research and development activities aimed at exploring the instructional implications of computer assistance to an

individualized school have focused on the mathematics and language arts curricula. As a further means of focusing project efforts, computer activities have concentrated on the intermediate wing at Oakleaf, comprising one third-grade, two fourth-grade, and two fifth-grade classrooms.

Summarized in this progress report are the project's accomplishments to date in each of the following areas: computer system design, computer-assisted testing (CAT), computer-assisted instruction in spelling, computer-assisted practice for mathematics skills, computer-assisted instruction in problem-solving techniques, and computer-assisted management of instruction. Plans for 1972-73 are also presented for each area. The final section of the report discusses issues that must be considered in determining the computer's potential as an integral and useful component of an adaptive, elementary-school environment.

COMPUTER SYSTEM DESIGN

Robert J. Fitzhugh

Work in the area of computer system design has been directed toward providing a research facility that can be utilized: (1) to determine the support that a computer can furnish to an individualized educational environment in the areas of management, testing, and instruction, and (2) to provide data and experience upon which alternative specifications can be developed for a small computer system which is within the economic reach of many school districts. Toward this end, an Educational Time-Sharing System (ETSS) has been developed, tested, and installed. ETSS is a general-purpose, multi-language, time-sharing system, implemented on a mini-computer (a 24K Digital Equipment Corporation PDP-15), but offering features, services, and interactive capabilities presently available only on larger computer time-sharing systems. The computer and associated hardware, originally installed at LRDC to facilitate software development, were moved to the Oakleaf School in January 1972 and installed in a specially modified Fruehauf van located to the rear of the school. The system was successfully time sharing within a week, and was made available to the user community on a half-day basis beginning in late January. In February, the system was made available on a full-day basis, and in March, the schedule was further expanded to a 12-hour operations day. Hardware and software reliability has been excellent, and actual up-time has consistently exceeded 99 percent of scheduled up-time. Mean-time-to-failure over the past seven months is 46.5 operating days.

The T-64 language processor (a language similar to Calvin Mooer's TRAC™) and DEBUG (an on-line assembly language debugging system) were installed as the first of a number of language SUB-SYSTEMS. Installed during the months of February and March were a LOADER Sub-System used to load and link binary relocatable FORTRAN IV or MACRO ASSEMBLER programs and FOCAL™, a JOSS-type conversational algebraic language developed by the Digital Equipment Corporation and modified by LRDC to operate under ETSS. The ETSS EDITOR, MACRO ASSEMBLER, and FORTRAN IV COMPILER Sub-Systems were installed in April. A DECTAPE FILE MANAGEMENT Sub-System was added in late August, bringing the total number of available SUB-SYSTEMS to eight.

The need to locate the ETSS system on-site at the Oakleaf School was an outgrowth of a desire to avoid teletypes and to seek out a faster and improved terminal device. At higher data rates, telephone line and equipment charges become substantial, making it most economic to locate on-site and avoid the telephone system entirely. Although the Datapoint 3300 cathode-ray tube (CRT) terminal currently in use is satisfactory, there is continued interest in enhanced terminal devices, particularly those with audio as well as video components. Programs requiring audio are forced to run on LRDC's existing PDP-7, which has a limited audio capability.

Currently, the ETSS system supports 32 terminals, 16 located in the school and 16 accessing the system on a telephone dial-up basis. The capability for remote access was added to permit programmers and researchers engaged in program development to work from LRDC rather than at the computer site.

ETSS software is fully documented in two parallel documentation series. The two-volume ETSS Internal Reference Specifications provides a detailed description of the internal operating characteristics of the time-sharing system. User programming information is found in the four-volume ETSS System Reference Manuals.

Although many of the characteristics of ETSS were implemented in direct response to application-specific requirements, ETSS is a general-purpose, time-sharing system rather than a special-purpose system designed only for an educational environment. This decision to develop a general-purpose system places LRDC at variance with several similar projects at other institutions and was based on a number of considerations. First, the system was developed primarily as a research venture rather than as a prototype of a system that might be replicated and produced commercially. As a research tool, the ability to meet changing requirements was essential. It was felt that the researcher should encounter as few artificial constraints as possible, even though the price of generality is often increased programming difficulty.

Second, the system must support several dissimilar applications. Although testing and instruction are nearly identical from a computer point of view, the data management and retrieval applications posed an entirely different set of requirements. Rather than attempting to define and isolate common functions that could be incorporated in a "special-purpose" system, the development of a general-purpose system seemed to be the cleanest solution.

Third, the development of a special-purpose system would most probably preclude the use of commonly available and widely

understood programming languages such as FORTRAN or BASIC™. A programming language or programming medium would have to be defined and implemented, and this step was felt to be distinctly non-trivial. In addition, there was a reluctance to contribute another unusual, one-of-a-kind language to the existing Tower of Babel, particularly in the absence of a clearer understanding of its potential applications. The development or, at least, the definition of an application-specific language will be more appropriate once substantial in-school experience has been acquired.

Finally, the decision to develop a general-purpose, small computer, time-sharing system was further reinforced by the belief that systems, much like ETSS, will be commercially available from computer manufacturers within several years. A number of systems are currently available that offer the language BASIC™ in a time-shared mode, and several multi-language systems are planned or under development. The marketing success experienced by companies such as Hewlett-Packard with their time-shared BASIC™ system indicates that future, more powerful systems such as ETSS will see wide use. For educational applications such as those planned for the Oakleaf School, a commercially available, small-machine system would seem to be both a likely and realistic future possibility when contrasted with either large, or small, special-purpose educational computer systems.

During the coming year, ETSS development will focus primarily on software rather than hardware. Any additional hardware that is acquired will be interfaced and installed by the manufacturer. In addition to the applications programming effort, which is described

elsewhere in this report, systems software development will continue in three primary areas. First, a number of time-sharing system developments are planned, including the implementation of the remaining MONITOR commands, several extensive EXECUTIVE modifications, and a number of extensions to the existing input/output structure. The major project in the second area of SUB-SYSTEM (language processors, utility systems) development will be the development of a LOGO language Sub-System. Work will begin in March 1973 with installation scheduled for September 1973.

The third area of software activity during the year will be concerned with monitoring and evaluating the performance of the system. This effort will go beyond merely monitoring performance and will address itself to the broader issue of determining the extent to which a small computer system is able to satisfy the computer requirements of an individualized elementary school. Over the next few years, the results of this work will provide a basis for the development of specifications for a multi-purpose, cost-effective, small computer system for elementary schools.

COMPUTER-ASSISTED TESTING

Tse-Chi Hsu and Marthena Carlson

As part of the Center's efforts to develop procedures for individualized instruction, a computer-assisted testing model has been developed for the Individually Prescribed Instruction (IPI) mathematics curriculum at Oakleaf as a means of investigating the potential of adaptive testing (Ferguson, 1971; Ferguson & Hsu, 1971). The advantages of using computer-assisted testing rather than paper-and-pencil tests can be summarized as follows:

1. CAT more readily permits branched testing according to the performance of the student.
2. When using CAT, it is easier to vary the number of test items according to the student's performance on the items. If he is consistently either wrong or right, a decision of non-mastery or mastery can be made earlier than would be possible with paper-and-pencil tests. If the pattern of responses is not in a sequence of all right or all wrong, more items can be administered. The additional items permit a more valid decision than is possible with paper-and-pencil tests.
3. The student's time spent in testing is reduced, due in part to the hierarchy which requires fewer objectives to be tested, and in part to the provision for an early decision about mastery and non-mastery as discussed in (2).

4. The computer can be used to generate any number of equivalent tests for evaluating a given skill or unit, eliminating the problems associated with using the same test more than once.

5. The computer scores the tests and provides an immediate summary of results in a more precise form than is possible with paper-and-pencil tests.

The present model of computer testing relies on a hierarchical structure for mathematical skills that defines prerequisite relationships among the objectives of a curriculum unit. Branching strategies are determined from the hierarchy and curriculum content for testing purposes. For each objective, several "item forms" are identified. Each item form consists of a domain of item types, with a list of generation rules that precisely define that domain of items. Each item form represents a task or behavior identified in an objective. Test items are generated from each item form as a student takes a test. At the conclusion of testing, a summary of the student's performance by item form is presented, and a teacher can prescribe appropriate instruction.

The CAT model consists of five basic components: (1) TESTING MANAGER, (2) PARAMETER CONTROLLER, (3) ITEM GENERATOR, (4) ITEM ADMINISTRATOR, and (5) DECISION MAKER. A detailed description of each of the components has been provided by Ferguson and Hsu (1971). According to this model, the TESTING MANAGER initiates testing with an item on an objective specified by the teacher. The particular item presented to the examinee is constructed by the ITEM GENERATOR in accordance with parameters

specified by the test builder by way of the PARAMETER CONTROLLER. After each item is presented and scored by the ITEM ADMINISTRATOR, the DECISION MAKER determines whether or not the examinee's proficiency status can be declared. Wald's Sequential Probability Ratio Test (Wald, 1947) is employed as the decision strategy. If the decision cannot be made, the TESTING MANAGER calls for the generation of another item on the same objective. If a decision can be made, the TESTING MANAGER assumes control and branches to test another objective. When all necessary testing has been completed, the MANAGER summarizes the student's performance and presents diagnostic information for use in prescribing new work. The same model is used for both pre- and posttests and curriculum-embedded tests (CETs). The only difference is that Wald's test is not used for CETs because an equal number of items of each item form is required to provide diagnostic information about a student's performance within an objective.

Based on this model, experimental programs were written and implemented at the Oakleaf School for two multiplication units of the intermediate mathematics curriculum. These units cover roughly third- and fourth-grade material, beginning with an introduction to multiplication and the multiplication of single digits and extending into word and computational problems involving two-digit numbers multiplied by numbers of three or more digits. More than one unit was selected for development so that it would be possible to study the problems encountered in modifying the programs for different content. New teaching sequences for the units were developed in conjunction with the computer programs for CAT. This, however, is not a necessary step. Computer-assisted testing programs can be developed for any existing

curriculum, provided the curriculum has a reasonably good hierarchical structure. There are certain advantages though in developing new instructional materials and testing programs at the same time. Testing hierarchies, branching sequences, and item forms for CAT can be used in designing instructional materials, and test results can be interpreted and utilized readily in instruction.

In order to understand how CAT fits into the school operation, it is necessary to review the current measurement procedures of IPI. Placement tests are usually administered at the beginning of the school year in order to determine the unit where instruction should begin. After the unit is identified, a pretest is administered to make sure that the student is really in need of instruction in that unit. If the student can pass the pretest, he can skip the unit and go on to the next higher one. If he fails all or part of the pretest, the teacher prescribes some learning materials for each skill that was not mastered. Two or more curriculum-embedded tests are provided at the end of the teaching sequences to determine whether the student's proficiency is adequate for that skill. Eighty-five percent correct is the arbitrary criterion for mastery. A posttest is administered when a student masters all skills in the unit. With the exception of initial placement, the CAT programs are designed to generate and manage all the IPI testing procedures. In addition, the programs can be used to generate exercise pages for practice problems or for test construction according to prespecified item forms--a function which teachers find very useful.

During the 1971-72 school year, the focus was on formative evaluation of the CAT model. Data obtained from field testing were used to improve the model, the computer program, teaching sequences,

and management procedures. A detailed evaluation report is currently in preparation. The modifications which substantially reduce testing time are the following: (1) adjusting the arbitrary criteria for mastery and non-mastery in Wald's probability ratio test so that a student is placed in the mastery or non-mastery group sooner; (2) reducing the maximum number of items administered for each skill; and (3) cumulating pre- and posttest item form data into CET files in order to avoid unnecessary duplication of testing. Other improvements include making it possible for requests for exercise pages for different skills to be made at one time, providing item form identification numbers for each test item, and writing large subroutines as several small subroutines in order to facilitate the debugging process. Also during the past year, procedures were explored for evaluating the quality of test items generated by the computer (Sebatane, 1972). In 1972-73, a new procedure will be added to the model which will permit the monitoring of item form statistics, thereby providing for evaluation and improvement of item forms on an immediate basis.

In addition to adapting CAT programs from the DEC-10 to the DEC-15 system, developmental work for the 1972-73 year will focus on addition/subtraction and division units of IPI math. Moreover, three major studies are projected to investigate the psychometric aspects of CAT. The first study will involve an empirical comparison of paper-and-pencil testing and computer-assisted testing. The comparison will be made in terms of the validity of the mastery and non-mastery decision, time spent in taking the test, usefulness of the test results for instruction, etc. In the second study, Wald's sequential test and other statistics such as Bayesian statistics will be compared in order to determine the most effective decision rules for CAT. The

final study will continue to explore and implement procedures for evaluating test items generated by the computer. Ways to estimate reliability when tests are criterion-referenced and have varying numbers of items will be investigated.

COMPUTER-ASSISTED INSTRUCTION IN SPELLING

Karen K. Block and Dorothea P. Simon

The rich heritage of the English language has resulted in many departures from a strict alphabetic transcription of the language. However, even though phoneme-to-grapheme mappings are not strictly one-to-one, phonemes still provide important clues to the pronunciation and spellings of words in American English, and the number of graphemic options for each phoneme is limited. Additional information for spelling comes from orthographic invariances that hold across sets of words that are semantically related; for example, the spelling of night clues tonight and benighted. Still further information is provided by rules governing the addition of structural endings to root words. Two CAI programs in spelling have been developed that emphasize these sources of information. One program is on spelling patterns for selected phonemes (SPELPAT), the other on word families (Structural Analysis).

The lesson rationale for SPELPAT relies heavily on the concept proposed by Simon and Simon (1972) that a speller uses stored visual recognition information to select the correct graphemic option from possible spelling alternatives. Following from this concept, the SPELPAT program is designed to give students practice in attending to the graphemic options employed in spelling target sounds in chosen words. This is accomplished by requiring the student to: (1) read a sentence and identify words containing the target sound, (2) count the ways in which the sound is spelled in selected words, and (3) sort the words under their spelling patterns for that sound.

It is hypothesized that this instructional activity provides the learner with information required to generate alternative spellings of the same sound, and to select from among these spellings the correct one for a target word. The latter are critical components of spelling performance, and because the program is oriented to teaching these basic components, it can be called a tutorial program (Block, 1971).

Choice of material for the SPELPAT program was dictated by the spelling series in use at the Oakleaf School (Kottmeyer & Ware, 1964). From the 30 lessons in Book 4 of the existing curriculum, five were selected for CAI development. For each lesson, all the words that the student was expected to learn, plus others that fit the same spelling patterns, were put into sentences to be used in the program.

Several features of the SPELPAT program are worth noting. First of all, the student is given immediate feedback--reinforcement for a correct response, information about the nature of his error if the response is incorrect. Secondly, wild guessing or boredom is prevented by a correction procedure which allows not more than two incorrect responses; that is, after two consecutive errors on the same item, corrective feedback is provided. Third, in the final display, the columns that the student has generated illustrate by contrast the various graphemic options for the target sound. A final display for a SPELPAT "sorting" task on the /long o/ sound can be seen on the following page. The target spelling words are listed on the left; the student sorts them into the columns on the right.

	⟨OE⟩	⟨O-E⟩	⟨OA⟩
JOE	JOE	RODE	ROAST
RODE	HOE	HOME	
HOME		BONE	
ROAST		ALONE	
BONE		BROKE	
ALONE		SPOKE	
HOE			
BROKE			
SPOKE			

A final and important feature of SPELPAT is that it is completely general so that any group of words that illustrate graphemic options and that can be worked into a few lines can be plugged in. Thus, the program can be adapted to any grade level.

As currently planned, a student will use the SPELPAT program in place of the regular spelling lessons on the units programmed. After completing a lesson, he will take the posttest from the regular curriculum for words in that unit. An 85 percent correct criterion on the posttest will determine whether the unit has been mastered or if further work is needed. Initial use of the program will be limited to one grade so that management procedures can be developed into a smoothly operating system and the load on the computer consoles estimated. Once this initial phase is completed, all of the teachers in the intermediate wing at Oakleaf will be trained in the use of SPELPAT. With respect to evaluation of the program, current plans call for testing the extent to which students acquire the concept of optional spelling patterns for a sound, and the extent to which exposure to visual displays of words with like-spelled parts increases retention.

Structural Analysis, the second CAI program emphasizing phonemic clues and orthographic invariances in semantically related words, provides practice on structural rules and teaches the spellings of affixations and root forms. For each structural rule, the student engages in drill and practice on a set of words that illustrate the rule. There are two item forms for each word taught, a root item form and a derived item form. Each time the student responds to an item, he is told whether or not his spelling is correct. If it is not correct, he is told to "try again" and the word and its item forms are saved for re-presentation at the end of the word list. If the spelling is incorrect on the second try, the correct spelling is given and a new word presented. Exercises on a series of words illustrating a rule are terminated when the student correctly spells both the root and derived forms of all words on the list.

For each word in a lesson on prefixes and suffixes, there are also root and derived forms, and the order of presentation and contingencies are similar to those found in structural rule lessons. However, to respond to affixation item forms, the student must use his reading comprehension skills in addition to his spelling skills, since the response words are not directly available from the affixation item forms as they are from the structural item forms. Affixation item forms can be analogies, series, antonyms, or derivatives. They explore a range of semantic relationships that can hold between words and that provide clues to spelling. By requiring the student to respond to semantic and syntactic context to access the affixed form to be spelled and then requiring him to spell it in that context, it is expected that he will enrich his network of relationships among words, that he will use these relationships to organize the words he

knows and use them to generate new words, and that he will make use of these relationships in spelling.

In the initial phase of implementation, all students who will be receiving spelling instruction via CAI will take the pretest from the regular curriculum on the units to be treated in CAI. The school pretest is given on cassette tapes: words are pronounced, used in a sentence and pronounced again, and the student writes the spelling. Students will not be given feedback on their pretest results before reporting to the computer terminals to work through the Structural Analysis program. Because the program cycles through the entire list of spelling words before re-presenting any words missed, the first "run" through the list serves as a second pretest. The relationship between these two forms of testing spelling will be investigated.

After each root and derived form has been spelled correctly, instruction on power words begins. Power words are representative of other curriculum areas, such as "Wisconsin" from geography, and do not necessarily illustrate the structural spelling rule taught in the immediately preceding lesson. Each spelling unit in the regular series has associated with it a set of power words, and a unit is not considered mastered until 85 percent of the words illustrating a rule and the power words are spelled correctly on a taped posttest. The student is not given feedback as to the results of the test until he takes a specially designed paper-and-pencil posttest that tests, in addition to spellings of words treated in instruction, other objectives of CAI such as skill in applying a structural rule to new words. Six weeks after posttesting, students receive special retention tests to find out what has been retained from the programmed lessons.

Evaluation of the Structural Analysis program will include: (1) a comparison of the two forms of testing--audio testing, where the word is pronounced, and non-audio testing, where the word to be spelled is prompted by the syntactics or semantics of a sentence or analog; and (2) assessment of the generality of spelling achievement to language usage that results from learning spelling in the context of affixation rules. As larger portions of the spelling curriculum are programmed, the evaluation of performance on standardized spelling tests will be in order.

In addition to the CAI programs which can replace regular school instruction, two other programs have been written and tested. One of these programs, Word Puzzles, is intended to serve as a maintenance activity in spelling and to explore the use of display features of the Datapoint CRT terminal and the current audio facility that is part of the Center's PDP 7/9 configuration. The program is centered around the spellings of the /sh/ and /ch/ sounds and the /ou/ sounds. A student is given one, two, or three clues as to the spelling of a word containing the target sound. He responds to the first clue; if he is correct, the word is placed in a design on the screen; if he is not correct, he gets one or two additional bits of information, or the target word. The clues can be audio or graphical or definitional. Word Puzzles is currently available for demonstration in the laboratory.

The second program is aimed at teaching contractions and the rule which governs them. The interesting point to this program is that as students respond to it, they generate a protocol of "contractions," some of which are allowable in English speaking and writing (e.g., Nancy + will = Nancy'll) and some of which are not (e.g.,

go + not = gon't). The computer, operating according to a rule, produces both kinds; another source of information must be consulted to determine which are allowable. When the program is implemented at Oakleaf, students will be instructed to consult their dictionaries and other sources to determine whether the contractions formed are ones that the English language permits.

Another major activity planned for the coming year is the development of a more comprehensive tutorial system for spelling. The features presently envisioned for such a program include: (1) a detailed answer-processing system capable of recognizing certain classes of errors; (2) a history function that accumulates error information and decides when remedial treatment should be given; and (3) a selection routine capable of using error information to select exercises and word lists appropriate to any spelling difficulty. These latter exercises might consist of auditory training, visual discrimination training such as the exercises in SPELPAT, review of a structural rule, review of graphemic options for a target sound, or spelling words in the same word family as the target word.

Since misspellings can arise from inadequate auditory analysis skills (spellings such as shurging for shrugging, and pajma for pajama illustrate this problem), it is planned that instructional modules for such skills will be explored. These modules will make use of rhyming skills, segmentation, etc., and will be tested on LRDC's current audio system. Questions will be asked concerning whether the current audio units can provide adequate access time and freedom from extraneous noise. This experience should provide data for determining specifications for a computer-controlled audio system for elementary-school instruction.

In addition to instruction devoted to correcting deficits in spelling power, a portion of instruction should be directed toward developing self-monitoring skills that alert the student to words with which he will have (or has had) difficulty and that lead him to check his spelling. To develop this ability in students, several tasks have been designed that require the student to make a prediction about whether or not he will spell a word correctly, or make a judgment about his previous spellings. Another task has been designed in which students "buy" portions of a word that might present them with some difficulty when they try to spell it. Over the next few years, tasks such as these will be implemented on the computer in a game-like context.

Future instructional development will also be devoted to exploring ways to develop children's facility with a "generate and test" strategy for learning spellings (Simon & Simon, 1972). Prototype lessons will be designed and tested that allow children to spell words in multiple ways and then learn the correct spellings in the set, and that teach self-monitoring skills in spelling (Block, Tucker, & Peskowitz, 1971).

COMPUTER-ASSISTED PRACTICE FOR MATHEMATICS SKILLS

Eric Jacobson

Practice is an instructional treatment which has been traditionally applied in the teaching of arithmetic routines. Initial instruction introduces a student to a routine and teaches him to perform it with reasonable accuracy. Subsequent practice consolidates the routine so that the student performs it quickly, easily, and accurately, and retains this ability for long periods. Practice has several properties which make it particularly suitable for computer presentation. First of all, non-computer presentation, preparation, and monitoring of practice problems require a great amount of clerical work--work that is often avoided by school personnel not only because it is time-consuming but also because of the tedium involved. Computer presentation permits school personnel to spend their time at more challenging and productive tasks. With a computer, it is also possible to give the student immediate error feedback on each problem. Without the computer, the student gets feedback only after his work has been checked by the teacher. Finally, the computer can produce problems to meet any specified requirements and can select specifications for the next problem immediately after completion of the last--a feature not feasible without computer assistance.

For these reasons, a program (Drill 2) was developed to enhance the learning and retention of arithmetic routines at the Oakleaf School and, in so doing, discover some of the variables which determine the effectiveness of computer-presented practice. An initial set of lessons was designed to provide practice in multiplication, and

to determine the proper temporal relationship between practice and the prior mastery of the practiced skill.

Since an individualized program such as the one operating at Oakleaf specifies each individual's instruction, it is possible to control precisely the amount of prior non-practice instruction each student has before he begins practice. This control of the individual curriculum makes it possible to conduct research into the relationship of practice and other instruction in the learning of a particular skill. Thus, an individualized program not only provides a setting in which a student can practice skills at that point in the curriculum where practice will be most effective, but also provides an opportunity to investigate the specific effects of practice in the context of well-controlled prior instruction.

The present practice program consists of problems taken from three multiplication units of the IPI mathematics curriculum. These units cover roughly the traditional third-, fourth-, and fifth-grade multiplication material, beginning with an introduction to multiplication and the multiplication of single digits and extending into the multiplication of four-digit mixed decimal numbers. The material from these units is divided into three categories--simple multiplication, operational associativity, and decimal multiplication. Within each of these categories, hierarchies of problem types (or item forms) are defined. A problem type defines limited properties of a problem within which the program selects specific values randomly. During program operation, problem types are selected by moving up or down the hierarchy when the student answers the previous problem correctly or incorrectly.

At present, students begin work on the practice program only after they have completed an IPI multiplication unit. Completion, as defined in the IPI program, constitutes passing a posttest with at least 85 percent of the answers correct. A student enters the practice hierarchy at the highest level of difficulty encountered in the unit just completed. This level is defined as the difficulty ceiling for that student, so that no problems are practiced which have not been mastered in the IPI curriculum. Thus, upon entering the program, the student may move down the difficulty hierarchy but not up, and if he does move down and then up, he cannot surpass this entering level. Later applications of the program may allow him to continue on beyond his level of classroom mastery so that instruction and practice might proceed together and complement one another.

Three independent difficulty hierarchies are used in the program. The simple multiplication hierarchy contains problems from all three IPI multiplication units. The operational associativity hierarchy pertains only to one unit, while the decimal multiplication hierarchy contains problems exclusively from another. For students in these two units, the class of problems is initially chosen by a weighted random selection. For students in all units, when the class of problems has been determined, the hierarchy for that class is examined and a problem type of the appropriate difficulty selected. A specific problem of that type is then generated and shown to the student. The student's answer is evaluated and his position on the hierarchy adjusted accordingly. When this cycle is complete, the problem class is again randomly selected.

The multiplication problems are presented on a Datapoint CRT terminal. The student reads the problem, computes the answer, and

types it in at the terminal. If the answer is correct, a speed score is computed using the formula,

Speed score = (mean latency) - (problem latency) + 30,

where mean latency is the mean answer latency in seconds for all problems completed correctly by the student in his third practice session, and problem latency is the answer latency in seconds for the particular problem under consideration. The purpose of the speed score is to motivate rapid computation. The faster the student answers, the higher the speed score. If the score is between 15 and 40, the message "correct" appears. If it is greater than 40, a positively affective reinforcement message is shown, such as "Nicely done" or "What a smarty you are." If the score is less than 15, a negatively affective message is presented, e.g., "You can go faster than that" or "Pretty slow for you." In the earlier practice sessions during which a speed score is unavailable, the student is simply given a positively affective message for any correct answer.

If the student's answer is wrong, he is given one of several possible messages depending on the type of error, then asked to try again. The same problem is printed under the error. If the student is correct on the second try, he is given a reinforcement message, there is a three-second wait, the screen is cleared, and a new problem appears. If he is wrong the second time, he is given the appropriate error message, the problem is given with the correct answer, and the following message is given, "Check your answer with the correct answer and press return when finished." When the return button is pressed, the screen is cleared and a new problem is selected and shown.

After the student completes the last problem of a practice lesson, he is given a closing message which contains the number of problems he has answered correctly and the total speed score for that session. The latencies for all responses in the session are recorded for analysis.

Two potentially productive ways in which the practice program can be used are being investigated: (1) practice as skill maintenance, and (2) practice as remediation. In practice as skill maintenance, the program is administered upon completion of an IPI unit, when a student shows mastery in all the objectives of that unit. Since mastery may not be retained after an interval of time and since it is difficult to predict what skills will be lost, practice given immediately upon mastery may be effective in providing overlearning so that the skills will be retained. On the other hand, such practice can be inefficient, since some students will be practicing skills which they would have retained without practice. Practice as remediation does not begin immediately after initial instruction, but is delayed for a certain period of time and used to reinstitute skills lost during the delay. In this way, only those students who lose skills will practice and they will practice only those skills which they have lost. This remediation can be more efficient than maintenance if the reacquisition of a forgotten skill is not a great deal more time-consuming than the maintenance of an acquired one. There are a number of questions about the effective use of practice involved here, and to explore these issues, experimentation is being conducted at Oakleaf.

Additional research and development activities planned for the coming year include an investigation of the unique advantages of computer-presented practice. The general strategy of this research

will be to develop two sets of practice lessons--one computer operated and the other pencil and paper. The paper-and-pencil practice will be constructed, as much as possible, to match in detail the computer practice. If computer practice does prove to be superior, it should be possible to analyze modes of presentation and to isolate the particular characteristics of computer presentation which underlie its greater effectiveness. Furthermore, by attempting to simulate with paper and pencil separate components of the computer program, information should be provided for determining the relative cost of different aspects of practice. One example of the type of outcome this research might lead to would be the finding that practice with branching diagnostic problem selection leads to significantly better retention than non-branching practice, but that daily branching decisions are as effective as branching decisions after each problem and that the former can be implemented least expensively in a paper-and-pencil system.

Another area of development which will be coordinated with work in the area of computer-assisted testing will place practice at the center of a curriculum to teach multiplication. Practice will not simply be a treatment aimed at insuring retention of a previously learned routine, but will also be used to initially teach the routine. Students will begin instruction in multiplication by beginning at the lowest difficulty level of the multiplication practice program, and will progress up the difficulty hierarchy until they fail to respond correctly. The program will then administer a special diagnostic test to determine the student's deficiency and on the basis of this information, will select a special teaching routine to overcome the particular difficulty encountered. Since success is immediately

rewarded and progress can be made easily apparent, such computer practice should provide a highly motivating circumstance for attending, working, and developing routine skills. In addition, the practice hierarchy is a sensitive diagnostic scale which can provide detailed information about what a student can and cannot do. For these reasons, a program such as the one which presently provides practice in multiplication could be the basis of highly efficient teaching programs. Development efforts for next year will consist of developing the missing components of this system, namely, the supplementary diagnostic tests and the teaching sequences.

Fundamental to arithmetic computation are the single-digit addition, subtraction, multiplication, and division operations called number facts. All arithmetic algorithms consist of reducing a problem to a set of single-digit operations, the results of which the problem-solver must supply from memory. The acquisition and overlearning of these number facts is thus a crucial step in arithmetic instruction.

Number facts acquisition resembles most closely paired-associate learning where the two integers and operation symbol serve as the stimulus and the result of the operation is the response supplied by the student. Recently studied properties of paired-associate learning can be used in the attempt to build an efficient and theoretically interesting computer program to teach number facts. One of these properties is the tendency for subjects to organize a set of paired-associates in the course of learning (Crothers & Suppes, 1967; Rotberg & Woolman, 1963). Sets of number facts can be presented initially to test the efficiency of different presentation orders, but ultimately to diagnostically select a particular presentation order

which is maximally beneficial for a student at a certain stage of learning. A second development in paired-associate learning is the demonstration that optimal item selection based on certain theoretical learning models can increase the efficiency of paired-associate acquisition (Atkinson & Paulson, 1970). The gist of this approach is that the item to be shown a student is the one which, on the basis of a theory of learning and the student's past performance, will show the greatest increment of learning due to presentation. After the completion of each item, a program can examine the list to select the next item on the basis of its optimal decision strategy. Work will continue during the coming project year on incorporating these two strategies, organization and optimal presentation, into a program to teach number facts.

COMPUTER-ASSISTED INSTRUCTION IN PROBLEM-SOLVING TECHNIQUES

Richard A. Roman

In addition to providing individualized instruction in specific subject-matter areas such as mathematics and spelling, the computer can also serve to individualize instruction in more generalizable skills such as problem solving. Students can exercise skills acquired through their regular curriculum by applying these skills to computer-presented problems requiring several logical steps to solution. In order to teach and study problem-solving skills, two programs have been developed: "Series" and "Functions." These programs provide problems in which the student's task is to describe the hidden regularities in a mathematical environment (although the programs are applicable to other content areas).

The components of problem-solving behavior emphasized are: (1) subdividing large problems into smaller ones, solving these, then integrating these solutions into a total solution; (2) formulating and testing hypotheses; and (3) deciding whether to continue working on a given path to solution or change to a new method. A further hypothesis involving this work is that in the initial learning of problem-solving procedures, externalization of the steps involved will enable the student to attend to his own thought processes and in this way lead to more general problem-solving behavior.

In the Series program, the student is presented with a problem in which he is asked to determine the next number or letter in a sequence. Under his control, the student may type in an input to test

his hypothesis and receive feedback about his responses, he may request presentation of additional data to gather more information, or he may terminate if he judges the task to be too easy or too difficult and go on to a new problem. Within the Series program, many types of problems are possible, extremely simple series (e.g., 5, 5, 5, 5, . . .) or series requiring the use of several principles at once (e.g., 6A, 7F, 8M, 9A, 6F, 7M, 8A, . . .). The Functions program selects and displays pairs of elements from a functional relationship. As in Series, the student has three options in making his next move.

Both programs provide a framework into which a variety of specific content can be placed. In the current implementation, different series and function types are used to follow up skills which have been learned in the IPI mathematics curriculum. In another, application of the program's specific objectives in mathematics can be taught by focusing problems on new skills which must be synthesized and discovered by the student. Within these programs, it is possible to draw on and branch to a wide range of mathematical sophistication appropriate to the student's level in the curriculum.

The coming project year will see a major formative evaluation and extension of the present problem-solving programs. At the moment, it is only a hypothesis that the programs will make students into more competent problem-solvers. It is already clear that students do become more competent within the context of the programs, but how much that skill transfers to other problems is still an open question. A characterization of "good" performers on each of the programs is required in order to design sequences of problems which lead students to behave more like the skilled performers. One approach

to this problem is through protocol analysis. Work has begun on tape recording students and skilled adults working on the Series program. It is now clear that certain specific concepts are missing from the repertoire of the poorer problem-solvers--concepts which could be taught by the program.

Other questions to be asked in formative evaluation include the following: Is there an important difference in time to mastery when a mathematical concept is taught by the Functions program and by the standard printed booklet plus manipulative materials? Is the Functions program faster or slower? If faster, is it faster enough to justify the cost? If slower, are the thinking skills learned important enough to justify the extra time? What are the long-term effects of working on these programs? Do the students maintain interest over a full year? Do they begin to get impatient with other modes of instruction?

A further question relates to the meaning of difficulty in the context of a sequence of problems. Examination of the Series program so far has shown that the ranked difficulty of the sequences of problems does not correspond with the performance of students. In ranking the sequences, it was assumed that all students would have acquired certain basic concepts for producing sequences, and that difficulty would depend on how many of these principles were combined and on the mathematical sophistication of the calculations. It turns out that many students are not able to use such basic principles as solving a sequence by examining two intertwined simple series (e.g., A, 1, B, 2, C, 3, . . .). The program is being redesigned to teach these principles.

During the coming year, procedures will also be investigated for the selection of problem sequences which challenge the student and systematically build his skill. The algorithm for selecting the next problem in a sequence is presently designed to slowly adjust the level of difficulty of problems as the student demonstrates that he is improving (or cannot yet handle) the level being given. The rate at which the algorithm changes difficulty is a critical variable for maintaining interest and the highest possible rate of improvement. A study will be made of alternative algorithms for this purpose.

A third type of exercise to teach problem-solving skills, in addition to the Series and Functions programs, is the student programming language LOGO developed at Bolt Beranek and Newman Inc. in conjunction with the Massachusetts Institute of Technology. In the initial teaching sequence used for LOGO, a student learns the fundamentals of making a computer do what he wants it to do. He learns procedures for writing, debugging, and editing programs; he learns the concepts of computer commands and names and their values; he learns to use previously written procedures as subprocedures, as well as to write simple recursive procedures; and he learns about infinite loops, and how to test a condition to terminate a procedure under some circumstances. LOGO was introduced into the Oakleaf School this past year; work will be continued on integrating it into the curriculum. In addition, the use of a newly designed mechanical "turtle" which can be programmed with the LOGO language will be investigated.

COMPUTER-ASSISTED MANAGEMENT OF INSTRUCTION

Gilbert Puente

Past efforts aimed at designing a computer-managed instruction (CMI) system for the Oakleaf School (see, for example, Cooley & Glaser, 1969) have indicated that an effective CMI system is unlikely to result in the absence of a detailed diagnostic testing program and a richer instructional program. This finding, coupled with the fact that a clerically operated system is adequate for current information requirements at Oakleaf, led to the decision to halt CMI design activities during the past year in favor of a review of existing procedures.

The introduction of computer testing and drill and practice in two mathematics units was observed and the manual procedures revised and augmented to accommodate these new instructional resources. A significant amount of effort was also devoted to training teachers and aides in managing the new environment and to eliciting suggestions for computer assistance from the Oakleaf Principal and School Coordinator. It seems clear that the impact of CAI and CAT will result in increasing demands for more timely and convenient access to student status and progress information, as well as a greater need for communication between teachers and aides since student learning will increasingly take place outside the teacher's view and control.

On the basis of this experience, an initial version of an on-line data-collection system for student progress data in mathematics has been designed and will be implemented in one classroom. The

system is presently designed for use by an aide, but will eventually be revised to enable students to enter their data as they complete various kinds of tests. The software for the system is being written in TRAC to permit flexible tryout of alternative data-acquisition procedures.

The objectives for CMI development in the coming year are: (1) to evolve a set of computer routines and concomitant procedures that will enable school personnel to cope with an educational environment relying to some extent on computer-assisted testing and instruction, and (2) to develop a system to assist the coordinator and teachers at Oakleaf in planning and subsequently monitoring the long-term instruction of individual students.

To accomplish the first of these objectives, three basic computer programs are planned, together with a data base of student mastery information. First, the present rudimentary data-collection programs will be extended to cover spelling and reading, as well as provide sufficient checking and correction features to enable students to enter their own data. A review and analysis of less extensive, reduced function CRTs will be made for possible use in this application, with a view toward installing such terminals in the classroom. Second, a series of computer-assisted monitoring procedures will be developed aimed at supplying teachers with individual student "status versus plan" information. These data can also be summarized and presented for use by aides in scheduling and monitoring the use of computer terminals. A system for maintaining and presenting such data will be developed and implemented.

The prime emphasis in CMI work during the coming year will be on the second of the above objectives, i.e., long-term planning.

The rationale for this work is that in individualized instruction, an important aspect of a teacher's management of instruction is the monitoring of each student's progress and the assignment of prescriptions. The present manual information system at Oakleaf provides adequate and timely information to teachers for performing this daily monitoring function. The present system, however, does not deal adequately with longer-term patterns of student performance. Long-term planning procedures could be improved if information were available which the manual system cannot now provide. In order to accomplish this, a goal-setting system will be designed to assist teachers in managing this aspect of instruction. The system will be experimental and designed to contribute answers to questions concerning, for example, the kinds of information that should be used by teachers in establishing a learning plan against which to monitor a student's progress, and the most appropriate form for displaying information to the teacher in order to maximize information transfer and minimize time to perform these functions.

USING A COMPUTER IN AN ELEMENTARY SCHOOL: QUESTIONS FOR STUDY

Jerome Rosner

LRDC's involvement at the Oakleaf School represents an attempt at developing an adaptive educational environment that is responsive to individual differences among children and teachers. A critical question, then, is: Can the computer serve as an integral and useful component of such an environment? The answer seems to depend upon three basic, interrelated, and equally important issues:

1. The teacher's perceptions and understandings of the computer as a readily managed instructional resource.

2. The effectiveness of CAI and CAT in functioning both as a support for existing instructional programs and as a vehicle for providing additional educational experiences that will stimulate intellectual growth.

3. The appropriateness of the interface between the elementary-school student and computer-assisted lessons and tests.

As far as the first of these issues is concerned, the computer will achieve maximum effectiveness only if it can be competently and comfortably used by teachers in a way that enhances their management of the total classroom. From the time CAI and CAT were introduced into the Oakleaf School, communications between developers and the school staff have been easy and effective. Teachers

have been kept well informed about various ways in which computer applications have been conceived, and their reactions and advice have been actively sought and incorporated in program development. Although the teachers seem to see advantages to computer support in an elementary-school program, their attitude, at the same time, is cautious. On the one hand, they seem to feel threatened by the computer; on the other, they appear to be unconvinced that it will help them in their tasks and concerned that it might impede the learning of some of "their children." In addition, CAI and CAT have thus far tended to complicate teacher management and decision making. To ease this situation, better management procedures are needed, along with a more flexible class scheduling system. (The current "block scheduling" approach has been shown to be a major source of difficulty.)

These concerns will probably pass, and the computer will assume more the role of a benign and useful servant. To facilitate this, two things will be done. First of all, in-service teacher training will be continued. Teachers will be given ample time and first-hand experience at the terminals so that they may become completely familiar with the variety of computer programs that have been prepared. It may also be advisable to instruct teachers in the basic rudiments of computer programming. In this way, they will begin to appreciate the limitations of the technology in its current state, and think more profitably in terms of how to exploit what is available. Secondly, plans are being made to make the terminals more readily accessible by moving them from their present location (a small conference room somewhat apart from the main instructional areas) to areas closer to the classrooms. Current plans also call for teaching

terminal users (teachers, aides, students) to manage the device without depending upon the assistance of supervisory personnel. (The terminals are now supervised by a trained person who initiates "LOGIN" procedures, calls for the proper program, etc.) To accomplish this, a "computer applications coordinator" will be introduced into the school for a limited period of time. This individual will be responsible for systematically eliciting and recording the comments of teachers, aides, and students regarding various aspects of applications. In addition, the coordinator will respond to teachers' inquiries concerning procedural and programming difficulties and, in general, attempt to facilitate integration of computer instruction into the school environment.

The second basic issue concerns the effectiveness of CAI and CAT in supporting currently available instructional programs, as well as in offering experiences that add to what already exists in the school. The former requires that computer-assisted lessons and tests be compatible with established school curricula. CAI material that is intended to teach objectives in existing school programs must use language and formats similar to those already in use; skills that are prerequisite to appropriate performance in computer lessons must be identified and attended to; and criteria for determining mastery of a computer lesson must be in accord with those applied in the classroom. The latter concern--that is, using the computer to provide an additional dimension to the school experience--could be extremely enriching, if it is developed effectively. One approach that has already been explored to some extent is teaching children to write their own programs using LOGO. This, as well as other approaches, will continue to be investigated.

The final issue that must be considered in determining the computer's potential as a resource in an elementary school concerns the appropriateness of the interface between the student and computer-assisted lessons and tests. So far, students have shown various reactions. Constraints on the nature of visual presentations, as dictated by currently available devices, cause certain problems for some children. A more flexible format would be useful; an audio component would be extremely helpful. The physical location of the terminals is yet another factor that probably affects children's attitudes. Some enjoy going to another room in the school, while others seem to be less secure about leaving the familiar territory of their classroom. (As mentioned above, the location of the terminals will be changed next year. They will be moved to "Learning Centers," rooms that will serve as instructional areas for a number of adjacent classrooms.)

Inasmuch as computer applications at Oakleaf are to be considered a prototype effort that will serve as a model for subsequent installations elsewhere, serious study of the problem of integrating the computer into an academic environment will be continued. A number of issues have been raised; a number of questions will be studied over the next few years. For example:

1. What is the most effective procedure for introducing computer applications to an elementary-school faculty and administration?
2. What level of technical awareness is needed by the teacher to optimize the implementation of computer instruction?

3. What prerequisite aptitudes should be manifested by students before exposure to CAI/CAT? Can these be taught or are they exclusively a function of maturation?

4. What motivational impact does CAI/CAT have on students ranging in age from six to eleven?

5. What is the preferred location for terminals, considering both teachers' and students' needs (e.g., in the classroom, conference room)?

6. What is the impact of the computer on an adaptive educational environment? How can computer instruction, testing, and management enhance the responsiveness of the system to the characteristics of each individual student?

7. What curriculum areas lend themselves to computer applications, given present technological constraints on display characteristics format?

8. Are there students who are now unable to interact profitably with CAI/CAT? How can the existing system be modified to accommodate them?

In balance, the favorable aspects resulting from installation of computer-assisted instruction and testing at Oakleaf outweigh the unfavorable. Clearly, CAI/CAT generated a certain amount of frustration and annoyance for all those involved--students, teachers, administrators (and, indeed, developers)--at the onset. However, as the hardware and software began to function more reliably and effectively, some hint of its ultimate potential became apparent. A great

deal of work remains, much development effort and ingenuity will be required, and many questions will have to be rigorously investigated. But, there is reason to believe that appropriate exploitation of available and yet-to-be-defined technology will provide educators with a powerful mechanism for effecting an instructional environment that is adaptive to individual differences in both students and teachers.

REFERENCES

Atkinson, R. C., & Paulson, J. A. An approach to the psychology of instruction. Stanford: Institute for Mathematical Studies in Social Science, Stanford University, 1970. (Technical Report 157)

Block, K. K. Strategies in computer-assisted instruction: A selective overview. Pittsburgh: Learning Research and Development Center, 1971. (Publication 1971/1)

Block, K. K., Tucker, S., & Peskowitz, N. B. Drill-and-practice in CAI spelling: Project interim report # 1. Pittsburgh: Learning Research and Development Center, 1971. (Publication 1971/16)

Cooley, W. W., & Glaser, R. The computer and individualized instruction. Science, 1969, 166, 574-582. (Also, LRDC Reprint 53)

Crothers, E., & Suppes, P. Experiments in second language learning. New York: Academic Press, 1967.

Ferguson, R. L. Computer assistance for individualizing measurement. Pittsburgh: Learning Research and Development Center, 1971. (Publication 1971/8)

Ferguson, R. L., & Hsu, T. The application of item generators for individualizing mathematics testing and instruction. Pittsburgh: Learning Research and Development Center, 1971. (Publication 1971/14)

Glaser, R., & Cooley, W. W. Instrumentation for teaching and instructional management. In R. M. W. Travers (Ed.), Handbook of research on teaching. (2nd ed.) New York: Rand McNally, in press.

Kottmeyer, W., & Ware, K. Basic goals in spelling. (2nd ed.) St. Louis: Webster Division, McGraw-Hill, 1964.

Rotberg, I. C., & Woolman, M. Verbal paired-associate learning as a function of grouping similar stimuli or responses. Journal of Experimental Psychology, 1963, 65, 47-51.

Sebatane, E. M. An empirical investigation of procedures for evaluating the quality of computer-generated tests. Unpublished Master's thesis, University of Pittsburgh, 1972.

Simon, H. A., & Simon, D. P. Alternative uses of phonemic information in spelling. Pittsburgh: Learning Research and Development Center, 1972. (Also, Review of Educational Research, 1973, in press)

Wald, A. Sequential analysis. New York: Wiley, 1947.